***CSS – by Hakon Wium Lie***

|  |  |
| --- | --- |
| ***HTML – by Tim Berners-Lee*** | HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two fundamental technologies used for building and styling websites. Together, they enable the creation of attractive and well-structured websites.  *Prepared By*  **PARANOIA TECHNOLOGIES** |

**HTML**

* **HTML** stands for **Hyper Text Markup Language**

"**Hypertext**" refers to the **hyperlinks that an HTML page may contain**. "Markup language" refers to the way tags are used to define the page layout and elements within the page.

**Difference between markup and programming language?**

* Markup languages are used for structuring and presenting information, while programming languages are designed for creating software applications and performing computational tasks.
* Markup languages focus on content presentation, while programming languages enable algorithmic and logical processing.
* Markup language is both human-readable and machine-readable while a programming language provides a set of commands and syntax that can be used to write computer programs which are understood by the computer.

**HTML**

* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
* The first version of HTML was written by **Tim Berners-Lee** in 1993.
* Since then, there have been many different versions of HTML. The most widely used version throughout the 2000's was HTML 4.01, which became an official standard in December 1999

**Tags**

* Special word or letter surrounded by angle brackets (<>).
* Types are paired and unpaired.

**Paired**

* Tag consists of an opening tag and closing tag.
* An HTML Paired tag starts with an opening tag: the tag name enclosed inside the angle brackets; for example, a paragraph opening tag is written as ‘<p>’.
* <p>This is a paragraph. </p>

**Unpaired**

* An HTML tag is called an unpaired tag when the tag only has an opening tag and does not have a closing tag or a companion tag.
* The Unpaired HTML tag does not require a closing tag; an opening tag is sufficient in this type.
* Unpaired tags are sometimes also named as **Standalone Tags or Singular Tags** since they do not require a companion tag.

**HTML Editors**

* Web pages can be created and modified by using professional HTML editors.
* However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).

**Structure**

<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
</body>  
</html>

* The **<!DOCTYPE html>** declaration defines that this document is an HTML5 document
* The **<html>** element is the root element of an HTML page
* The **<head>** element contains meta information about the HTML page
* The **<title>** element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The **<body>** element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The **<h1>** element defines a large heading
* The **<p>** element defines a paragraph

**HTML Elements**

* The HTML **element** is everything from the start tag to the end tag:

**Syntax:**

**<tagname>Content goes here...</tagname>**

## Nested HTML Elements

* HTML elements can be nested (this means that elements can contain other elements).
* All HTML documents consist of nested HTML elements.
* The following example contains four HTML elements (<html>, <body>, <h1> and <p>)

# HTML Headings

* HTML headings are titles or subtitles that you want to display on a webpage.
* HTML headings are defined with the <h1> to <h6> tags.
* <h1> defines the most important heading. <h6> defines the least important heading.

# HTML Paragraphs

* A paragraph always starts on a new line, and is usually a block of text.

# HTML Attributes

* All HTML elements can have **attributes**
* Attributes provide **additional information** about elements
* Attributes are always specified in **the start tag**
* Attributes usually come in name/value pairs like: **name="value"**

# HTML Images

* Images can improve the design and the appearance of a web page.
* The HTML **<img>** tag is used to embed an image in a web page.
* Images are not technically inserted into a web page; images are linked to web pages.
* The <img> tag creates a holding space for the referenced image.
* The <img> tag is empty, it contains attributes only, and does not have a closing tag.

**The <img> tag has two required attributes:**

* src - Specifies the path to the image
* alt - Specifies an alternate text for the image

### Syntax

**<img src="*url*" alt="*alternatetext*">**

# HTML Links

* Links are found in nearly all web pages.
* Links allow users to click their way from page to page.
* HTML links are hyperlinks.
* You can click on a link and jump to another document.
* When you move the mouse over a link, the mouse arrow will turn into a little hand.

## HTML Links - Syntax

* The **<a>** tag in HTML is used to create hyperlinks, which allow users to navigate between different web pages or sections within a webpage.
* The **<a>** tag stands for **"anchor"** and is often referred to as the anchor tag.

**<a href="*url*">*link text*</a>**

* The most important attribute of the <a> element is the href attribute, which indicates the link's destination.
* The ***link text*** is the part that will be visible to the reader.
* Clicking on the link text, will send the reader to the specified URL address.

By default, links will appear as follows in all browsers:

* An **unvisited link** is underlined and blue
* A **visited link** is underlined and purple
* An **active link** is underlined and red

**HTML Text Formatting**

HTML contains several elements for defining text with a special meaning.

* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Smaller text
* <del> - Deleted text
* <ins> - Inserted text
* <sub> - Subscript text
* <sup> - Superscript text

**HTML Styles**

* The HTML style attribute is used to add styles to an element, such as color, font, size, and more.
* Setting the style of an HTML element, can be done with the style attribute.

**Syntax:**

**<*tagname* style="*property* :*value;*">**

* The ***property*** is a CSS property. The ***value*** is a CSS value.

**CSS**

* **CSS** stands for **Cascading Style Sheets.**
* CSS was first proposed by **Hakon Wium Lie** on October 10, 1994. At the time, Lie was working with Tim Berners-Lee (father of Html) at CERN. The European Organization for Nuclear Research is known as CERN. **Hakon wium lie** is know as **Father of CSS**.
* CSS was proposed in 1994 as a web styling language, to solve some of the problems of Html 4. There were other styling languages proposed at this time, such as Style Sheets for Html and JSSS but CSS won.
* CSS saves a lot of work. It can control the layout of multiple web pages all at once.
* Cascading Style Sheets (CSS) is used to format the layout of a webpage.
* With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

## Using CSS

CSS can be added to HTML documents in 3 ways:

* **Inline** - by using the style attribute inside HTML elements
* **Internal** - by using a <style> element in the <head> section
* **External** - by using a <link> element to link to an external CSS file

## Inline CSS

* An inline CSS is used to apply a unique style to a single HTML element.
* An inline CSS uses the style attribute of an HTML element.

## Internal CSS

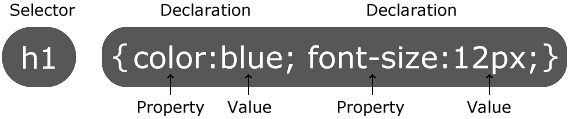
* An internal CSS is used to define a style for a single HTML page.
* An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

## External CSS

* An external style sheet is used to define the style for many HTML pages.
* To use an external style sheet, add a link to it in the <head> section of each HTML page
* The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

**A CSS rule consists of a selector and a declaration block.**

## CSS Syntax



* The selector points to the HTML element you want to style.
* The declaration block contains one or more declarations separated by semicolons.
* Each declaration includes a CSS property name and a value, separated by a colon.
* Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

**CSS Borders**

* The CSS border properties allow you to specify the style, width, and color of an element's border.

## CSS Border Style

* The border-style property specifies what kind of border to display.

The following values are allowed:

* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

**CSS Border Width**

* The border-width property specifies the width of the four borders.
* The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border)

**CSS Border Color**

* The border-color property is used to set the color of the four borders.

The color can be set by:

* name - specify a color name, like "red"
* HEX - specify a HEX value, like "#ff0000"
* RGB - specify a RGB value, like "rgb(255,0,0)"
* HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
* transparent

## CSS Border - Individual Sides

In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left):

If the border-style property has four values:

* **border-style: dotted solid double dashed;**
  + top border is dotted
  + right border is solid
  + bottom border is double
  + left border is dashed

If the border-style property has three values:

* **border-style: dotted solid double;**
  + top border is dotted
  + right and left borders are solid
  + bottom border is double

If the border-style property has two values:

* **border-style: dotted solid;**
  + top and bottom borders are dotted
  + right and left borders are solid

If the border-style property has one value:

* **border-style: dotted;**
  + all four borders are dotted

## CSS Border - Shorthand Property

* To shorten the code, it is also possible to specify all the individual border properties in one property.

The border property is a shorthand property for the following individual border properties:

* border-width
* border-style (required)
* border-color

# CSS Rounded Borders

The border-radius property is used to add rounded borders to an element:

**HTML Tables**

* HTML tables allow web developers to arrange data into rows and columns
* A table in HTML consists of table cells inside rows and columns
* Each table cell is defined by a <td> and a </td> tag.

## How To Add a Border

* When you add a border to a table, you also add borders around each table cell.
* To add a border, use the CSS border property on table, th, and td elements

## Collapsed Table Borders

* To avoid having double borders set the CSS border-collapse property to collapse.

## Round Table Borders

* With the border-radius property, the borders get rounded corners

## Dotted Table Borders

* With the border-style property, you can set the appereance of the border.

The following values are allowed:

* dotted
* dashed
* solid
* double
* groove
* ridge
* inset
* outset
* none
* hidden

## HTML Table Width

* To set the width of a table, add the style attribute to the <table> element

## HTML Table Column Width

* To set the size of a specific column, add the style attribute on a <th> or <td> element

# HTML Table Padding & Spacing

## HTML Table - Cell Padding

* Cell padding is the space between the cell edges and the cell content.
* By default the padding is set to 0.
* To add padding on table cells, use the CSS padding property

## HTML Table - Cell Spacing

* Cell spacing is the space between each cell.
* By default the space is set to 2 pixels.
* To change the space between table cells, use the CSS border-spacing property on the table element

## HTML Table - Colspan

* To make a cell span over multiple columns, use the colspan attribute

## HTML Table - Rowspan

* To make a cell span over multiple rows, use the rowspan attribute

**HTML <span> Tag**

* The <span> tag is an inline container used to mark up a part of a text, or a part of a document.
* The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
* The <span> tag is much like the [<div>](https://www.w3schools.com/tags/tag_div.asp) element, but <div> is a block-level element and <span> is an inline element.

**HTML Lists**

## Unordered HTML List

* An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

## Ordered HTML List

* An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

## Description HTML Lists

* HTML also supports description lists.
* A description list is a list of terms, with a description of each term.
* The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term

**CSS Selectors**

## The CSS id Selector

* The id selector uses the id attribute of an HTML element to select a specific element.
* The id of an element is unique within a page, so the id selector is used to select one unique element!
* To select an element with a specific id, write a hash (#) character, followed by the id of the element.

## The CSS class Selector

* The class selector selects HTML elements with a specific class attribute.
* To select elements with a specific class, write a period (.) character, followed by the class name.

## CSS Margins

* The CSS margin properties are used to create space around elements, outside of any defined borders.
* With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

## Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

* margin-top
* margin-right
* margin-bottom
* margin-left

**Position**

* The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

## The position Property

* The position property specifies the type of positioning method used for an element.

There are five different position values:

* **static**
* **relative**
* **fixed**
* **absolute**
* **sticky**

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

**HTML Forms**

## The <form> Element

* The HTML <form> element is used to create an HTML form for user input:

## The <input> Element

* The HTML <input> element is the most used form element.
* An <input> element can be displayed in many ways, depending on the type attribute.

## The <label> Element

* The <label> tag defines a label for many form elements.
* The **for** attribute of the <label> tag should be equal to the **id** attribute of the <input> element to bind them together.

**HTML Form Attributes**

## Action, Target, Method, Autocomplete, Novalidate.

## The Target Attribute

The **target** attribute specifies where to display the response that is received after submitting the form.

The **target**attribute can have one of the following values:

* **\_blank** – The response is displayed in a new window or tab
* **\_self** - The response is displayed in current window
* **\_parent** - The response is displayed in parent frame
* **\_top** – The response is displayed in the full body of window

## The Method Attribute

**Notes on GET:**

* Appends the form data to the URL, in name/value pairs
* NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
* The length of a URL is limited (2048 characters)
* Useful for form submissions where a user wants to bookmark the result
* GET is good for non-secure data, like query strings in Google

**Notes on POST:**

* Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
* POST has no size limitations, and can be used to send large amounts of data.
* Form submissions with POST cannot be bookmarked

## The Autocomplete Attribute

* The autocomplete attribute specifies whether a form should have autocomplete on or off.
* When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

## The Novalidate Attribute

* The novalidate attribute is a Boolean attribute.
* When present, it specifies that the form-data (input) should not be validated when submitted.

**HTML Form Elements**

* <input>
* <label>
* <select>
* <textarea>
* <button>
* <fieldset>
* <legend>
* <datalist>
* <output>
* <option>
* <optgroup>

**HTML Input Types**

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">

**HTML Input Attributes**

## value , readonly, pattern, placeholder, required

## CSS Flexbox

# CSS Flexbox (Flexible Box) is a layout model that allows you to create flexible and responsive designs for elements within a container.

* It provides a more efficient way to arrange, align, and distribute space among items in a container, whether they are in a row or a column.
* To use Flexbox, you first need to define a flex container by applying the **display: flex** property to its parent element.
* This will make all its direct children into flex items. Then, you can use various properties to control the layout and alignment of the items within the container.

**Here are some key properties used in Flexbox:**

* **display: flex:** This property is applied to the container to create a flex context.
* **flex-direction:** Specifies the direction of the main axis, which can be either row (left to right) or column (top to bottom).
* **flex-wrap:** Defines how the items should wrap if they exceed the width or height of the container. It can be set to nowrap, wrap, or wrap-reverse.
* **justify-content:** Aligns the flex items along the main axis. It defines the distribution of space between the items. Possible values include flex-start, flex-end, center, space-between, space-around, and space-evenly.
* **align-items:** Aligns the flex items along the cross axis (perpendicular to the main axis). Common values are flex-start, flex-end, center, baseline, and stretch.
* **align-content:** Similar to align-items, but this property is used when there is extra space in the cross axis. It applies to multiple lines of flex items and supports values like flex-start, flex-end, center, space-between, space-around, and stretch.
* **flex**: This property is used on individual flex items to specify their ability to grow, shrink, and their initial size. It combines three properties: **flex-grow**, **flex-shrink**, and **flex-basis**.
* **order**: Allows you to control the order in which flex items appear within the container. By default, all items have an order of 0, but you can assign positive or negative values to reorder them.

**Note:** Flexbox is particularly useful for creating responsive layouts that adjust based on screen size or device orientation. It simplifies complex layouts and reduces the need for floats or positioning.

**Example**

**HTML:**

<div class="flex-container">

<div class="flex-item">Item 1</div>

<div class="flex-item">Item 2</div>

<div class="flex-item">Item 3</div>

</div>

**CSS:**

. flex-container {

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

}

.flex-item {

flex: 1;

padding: 10px;

border: 1px solid #ccc;

}

**CSS Grid**

CSS Grid is a layout system that allows you to create two-dimensional grids of HTML elements. It is a powerful tool that can be used to create a variety of layouts, including:

* Responsive layouts that adapt to different screen sizes
* Complex layouts with multiple columns and rows
* Layouts with a fixed or fluid width
* Layouts with gutters or margins between columns and rows

Here are some of the **benefits** of using CSS Grid:

* It is a very flexible layout system that can be used to create a variety of layouts.
* It is very responsive, so your layouts will automatically adapt to different screen sizes.
* It is easy to use, even for beginners.
* It is supported by all major browsers.

Here are some of the properties that you can use to control the layout of CSS Grid elements:

**display:** This property specifies that the element should be a grid container.

**grid-template-columns:** This property specifies the number and width of the columns in the grid.

**grid-template-rows:** This property specifies the number and height of the rows in the grid.

**grid-column:** This property specifies the column in which an element should be placed.

**grid-row:** This property specifies the row in which an element should be placed.

**grid-gap:** This property specifies the amount of space between columns and rows.

**Example:**

**HTML:**

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

**CSS**

.container {

display: grid;

grid-template-columns: repeat(3, 1fr);

grid-template-rows: 1fr;

}

.item {

width: 100px;

height: 100px;

background-color: red;

}

This code will create a layout with three items that are evenly spaced in a row. If the screen size is small, the items will wrap to the next column.